

**Air Quality Index Prediction**

High Level Design

Domain: Machine Learning

Creator: Shridatta Patil

Date: 25.10.2023

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date issued | Version | Description | Author |
| Oct 25, 2023 | 1.1 | First Draft | Shridatta Patil |
|  |  |  |  |
|  |  |  |  |

Contents

[Abstract 3](#_Toc110595993)

[Introduction 3](#_Toc110595994)

[What is High-Level Design Document? 3](#_Toc110595995)

[Scope 4](#_Toc110595996)

[General Description 4](#_Toc110595997)

[Definitions 4](#_Toc110595998)

[Product Description 4](#_Toc110595999)

[Problem Statement 4](#_Toc110596000)

[Proposed solution 5](#_Toc110596001)

[Further improvements 5](#_Toc110596002)

[Data requirements 5](#_Toc110596003)

[Tools used 5](#_Toc110596004)

[Hardware Requirements 7](#_Toc110596005)

[Constraints 7](#_Toc110596006)

[Assumptions 7](#_Toc110596007)

[Design Details 7](#_Toc110596008)

[Process Flow 7](#_Toc110596009)

[Event Log 8](#_Toc110596010)

[Error Handling 9](#_Toc110596011)

[Performance 9](#_Toc110596012)

[Reusability 9](#_Toc110596013)

[Application Compatibility 9](#_Toc110596014)

[Resource Utilization 9](#_Toc110596015)

[Dashboards 10](#_Toc110596016)

[Conclusion 10](#_Toc110596018)

# Abstract

Air quality index is a tool for identify the present scenario of air quality. Six different

methods of estimating AQI bases on pollutants synergistic effect. The growth of

population, traffic and industrialization make it worse day by day. We to act in such

a way as to protect people.

The goal of this project is to build a prediction model using multiple machine learning

techniques and to use a template to document the end-to-end stages. We're trying to

forecast the value of a continuous variable with the AQI dataset, which is a regression

issue.

# Introduction

## What is High-Level Design Document?

The goal of this HLD or a high-level design document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of design aspects and define them in detail
* Describe all user interfaces being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and architecture of the project
* List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability

## Scope

The HLD documentation presents the structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

# General Description

## Definitions

|  |  |
| --- | --- |
| Term | Description |
| AQI | Air Quality Index |
| Database | Collection of the Information |
| Cloud | A data center full of services connected to the internet performing service |
| IDE | Integrated Development Environment |
| Heroku | A cloud service |

## Product Description

AQI is a Machine Learning based regression model which helps us to do predictive analysis of the quality of air using certain features and parameters.

## Problem Statement

To create an ML based solution for predictive analysis of quality of air and also deploy it in the form of a UI.

The aim to predict the air quality index. This is basically regression problem.

## Proposed solution

Using all the standard techniques used in the life cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a frontend where a user can fill their information in the form input and get the output instantly.

## Data requirements

Data requirement completely depend on our problem statement. We need the dataset from Pollution Control Department to improve accuracy of the model. Required dataset should contain the following features:

* PM2.5, PM10
* NO, NO2, NH3, NOx
* CO, O3, SO2
* Benzene, Toluene, Xylene

These are the required parameters to feed into model.

## Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask and a few other libraries were used to build the whole model.











* For visualization tasks, matplotlib, seaborn were used
* Flask is used for building the web application and server to run the code
* GitHub is used as version control system
* NumPy and Pandas were used to clean and interpret data
* Scikit-learn was used to cross validate and compare different models
* Random Forest Regressor was used to build the final model

## Hardware Requirements

* Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 512 MB of RAM
* 4 GB of hard-disk space

## Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

## Assumptions

The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible

# Design Details

## Process Flow

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

**Data Preparation**

**Model**

**Development**

**Deployment**

**Deployment**

## Event Log

The system should log every event so that the user will know what process is running internally. Initial step-by-step description: 1. The system identifies at what level logging is required 2. The system should be able to log each and every system flow 3. Developer can choose logging method. You can choose database logging/ File logging as well 4. System should not hang even after so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

## Error Handling

Errors should be encountered; an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal intended usage.

# Performance

The AQI tool is used to predict air quality index is bad or good by providing numeric value. It can be used by various governmental/ non-governmental/ private agencies then it is supposed to be as accurate as possible.

## Reusability

The code written and the components used should have the ability to be reused with no problems.

## Application Compatibility

The different components for this project will be using Python as an interface between them, each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

## Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished.

# Dashboards

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.



# Conclusion

All in all, overall project architecture, design details, used technologies and performance were explained in detail. The AQI will give the air quality index predictions instantly and has the potential to help various government organizations, agencies and etc.